EMBEDDED TECHNOLOGY & IOT Lab

(BEE01T1004)

Lab Manual

School of Electrical, Electronics and Communication Engineering

In-Charge	Name
Dean	Prof. B. Mahopatra
Programme Chair	Dr. Yogesh Kumar/Dr. Lokesh Varshney
Faculty members	ALL Faculty Members

University Vision

"To be known for world-class education, cutting-edge research, innovation, and application of knowledge to benefit society."

University Mission

M1: To provide high-quality education, knowledge and skills necessary for our students to be successful in the technologically evolving world.

M2: To provide a supportive learning environment that facilitates discovery of new knowledge and continuous innovation

M3: To instil a culture of interdisciplinary enquiry and education that facilitates generation of cutting-edge solutions to real-world problems.

M4: To foster an environment that inculcates skills in life-long learning and team based problem solving.

Department Vision

"To be known globally as a premier department of Electronics and Communication Engineering for value-based education and interdisciplinary research for innovation."

Department Mission

SM1: Create a strong foundation on fundamentals of Electronics and Communication Engineering through Outcome Based Learning Teaching (OBLT) Process.

SM2: Establish state-of -the-art facilities for design and simulation.

SM3: Provide opportunities to students to work on real world problems and develop sustainable ethical solutions.

SM4: Involve the students in group activities, including those of professional bodies to develop leadership and communication skills.

Programme Outcome (PO)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OBJECTIVES

- To provide the awareness of major embedded devices and interfacing devices
- To understand key technologies in Internet of Things.
- To analyze, design or develop parts of an Internet of Things solution for IoT applications.

COURSE OUTCOMES

On completion of this course, the students will be able to

CO1: Understand the concept of Arduino uno, Proteous s/w and their interactions.
CO2: Recognize and analyze given embedded system design and its performance.
CO3: Identify the programming environment to develop embedded solutions.
CO4: Demonstrate application-based competencies in Embedded Programming
CO5: Identify and adopt knowledge of the terminology, requirements and constraints for IoT system development.
CO6: Demonstrate IoT system for smaller applications

CO-PO Mapping

EMBEDDED TECHNOLOGY & IOT Lab (BEE01T1004)		Engineering Knowledge	Problem analysis	Design/development of solutions	Conduct investigations of complex problems	Modern tool usage	The engineer and society	Environment and sustainability	Ethics	Individual or team work	Communication	Project management and finance	Life-long Learning	PSO 1	PSO 2	PSO 3
COs		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Understand the concept of Arduino uno, Proteous s/w and their interactions	3	2	2						2			2			
2	Recognize and analyze given embedded system design and its performance.	3	2	2												
3	Identify the programming environment to develop embedded solutions.	3	2	1						2			2			
4	Demonstrate application based competencies in Embedded Programming	3	2	2						2			2			
5	Identify and adopt knowledge of the terminology, requirements and constraints for IoT system development.	3	2	1						2			1			
6	Demonstrate IoT system for smaller applications	2	2													

Mode of Evaluation

Components	Laboratory	
	Internal Examination	End Term Examination
Marks	25	25
Total Marks	50	

Details Evaluation Scheme:

Component of evaluation	Evaluation	Rubric for CO	Marks
Experiment understanding	**Internal**	Design process	5
Performance		Discussion of results	10
Record		Quality of sketch, drawing and graphs	5
Internal viva		Theory, tools, &team works	5 (1+2+2)
Lab experiment	**End term**	Design process	10
Lab Report		Quality of sketch, drawing and graphs	10
Viva by external expert		Theory, tools, &team works	5(1+2+2)
Total			50

ET&IOT LAB Lab Assessment Process-

- Faculty members must carry the attendance register.
- Before coming to class faculty members must have ensured the students get the lab manual.
- Clearly define the lab problem to the students and the expected outcome of the experiment.
- Clearly explain the objective and theory behind the lab experiments.
- All faculty members in a lab class shall actively participate in the lab experiment giving guidance to students.
- Faculty members must check the results obtained by each student and sign on it.
- Faculty members must correct the error in results and instruct student to do necessary modification in experiment to get the correct results.
- Faculty must take a note of any mal functioning of equipment or component if found during the tour of lab.
- Faculty must check and correct the student's lab records.
- Faculty members evaluate the student's performance in the lab class as a part of continuous evaluation.
- Faculty must give the assignment or lab problem to students for lab based solutions and shall assess the course outcomes based on performance of students.
- Faculty must ensure that each student endorse the following and upload in Moodle;
    - ✓ Preparation of data table and plot the graphs
    - ✓ students must explain data in table or graphs
    - ✓ Students must write the observation on data pattern or behavior of graphs.
    - ✓ Students must write the scientific justification of data variation or graphs behavior.
    - ✓ Students must write the error in results if any obtained during experiment.
- One course file is to be maintained for each course and all faculties must put the necessary documents of practice in the course file time to time.
- Faculty must declare the title of next experiment and must the students to go through lab manual before coming to lab.

LIST OF EXPERIMENTS

ATTEMPT ANY 6 EXPERIMENT FROM GIVEN LIST:

1. Study of Arduino Architecture and PIN details
2. Introduction of Arduino IDE and Proteus
3. Basic experimentation using LEDs
4. Interfacing IR Sensor with Arduino
5. Interfacing Ultrasonic Sensor with Arduino
6. Interfacing DC Motor with Arduino
7. Design of a simple home automation system
8. Design of a Traffic Light Controller
9. Design of Obstacle Detector using IR and Ultrasonic sensors
10. Design of a Line Follower Robot using Arduino
11. Design of a Bluetooth based system (Can give this as Additional Experiment)

Experiment 1:

Aim: Study of Arduino Architecture and PIN details

Theory

Component Explanations

Analog input pins – pins (A0-A5) that take-in analog values to be converted to be represented with a number range 0-1023 through an Analog to Digital Converter (ADC).

ATmega328 chip – 8-bit microcontroller that processes the sketch you programmed.

Built-in LED – an on board LED to pin 13.

Crystal Oscillator – clock that has a frequency of 16MHz

DC Jack – where the power source (AC-to-DC adapter or battery) should be connected. It is limited to input values between 6-20V but recommended to be around 7-12V.

Digital I/O pins – input and output pins (0-13) of which 6 of them (3, 5, 6, 9, 10 and 11) also provide PWM (Pulse Width Modulated) output by using the analogWrite() function. Pins (0 (RX) and 1 (TX)) are also used to transmit and receive serial data.

ICSP Header – pins for "In-Circuit Serial Programming" which is another method of programming.

ON indicator – LED that lights up when the board is connected to a power source.

Power Pins – pins that can be used to supply a circuit with values VIN (voltage from DC Jack), 3.3V and 5V. 2

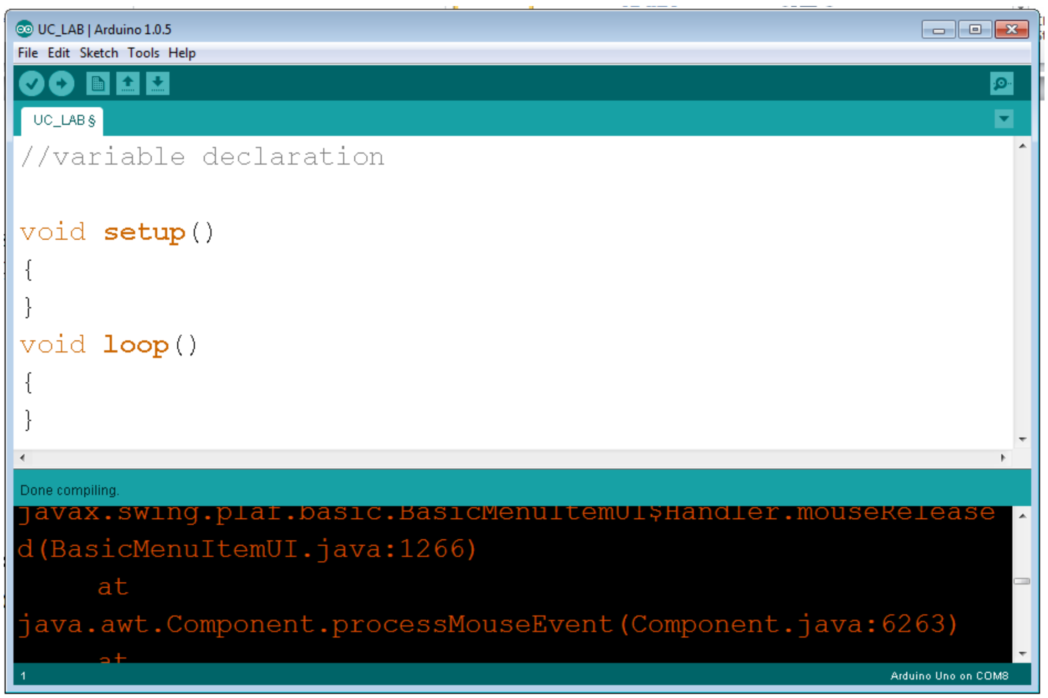
Reset Button – a button that is pressed whenever you need to restart the sketch programmed in the board.
USB port – allows the user to connect with a USB cable the board to a PC to upload sketches or provide a voltage supply to the board. This is also used for serial communication through the serial monitor from the Arduino software.

Experiment 2:

Aim: Introduction of Arduino IDE and Proteus

Theory:

Integrated Development Environment (IDE):
Once the board is installed, it's time to open the Arduino IDE.
It is fairly simple when compared to larger desktop C language development systems.



Experiment 3:

Aim: To start off, we will work on blinking an LED.

Theory and Program

LEDs are small, powerful lights that are used in many different applications. That's right - it's as simple as turning a light on and off. It might not seem like much, but establishing this important baseline will give you a solid foundation as we work toward more complex experiments.

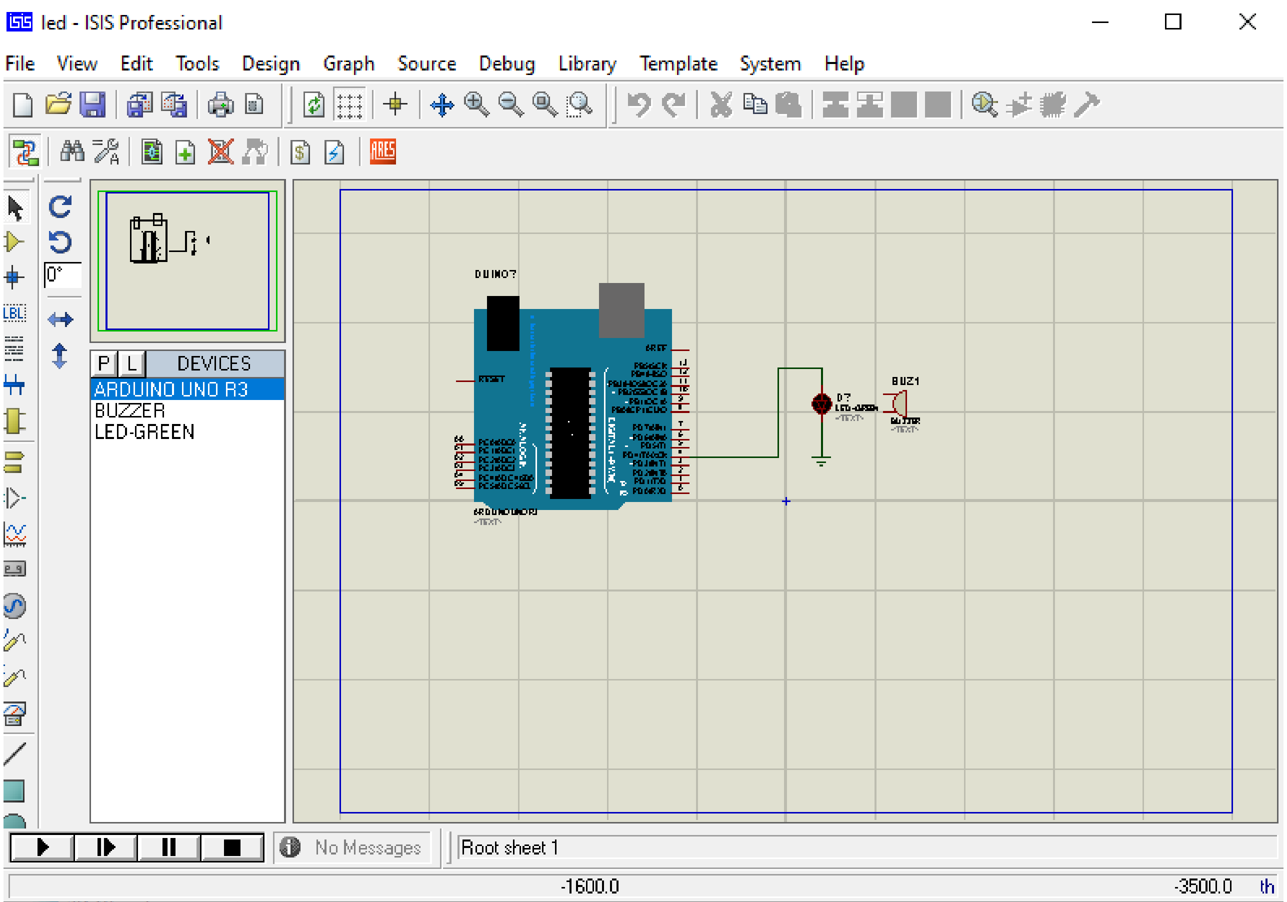
Required Equipment:

A. **1x** Breadboard
B. **1x** RedBoard or Arduino Uno
C. **1x** LED
D. **1x** 330Ω Resistor
E. **2x** Jumper Wires

Program-1 led blinking

```
int led =5;

void setup()

{

 pinMode  (led,OUTPUT );

}

void loop()

{

 digitalWrite(led, HIGH);

 delay(200);
```

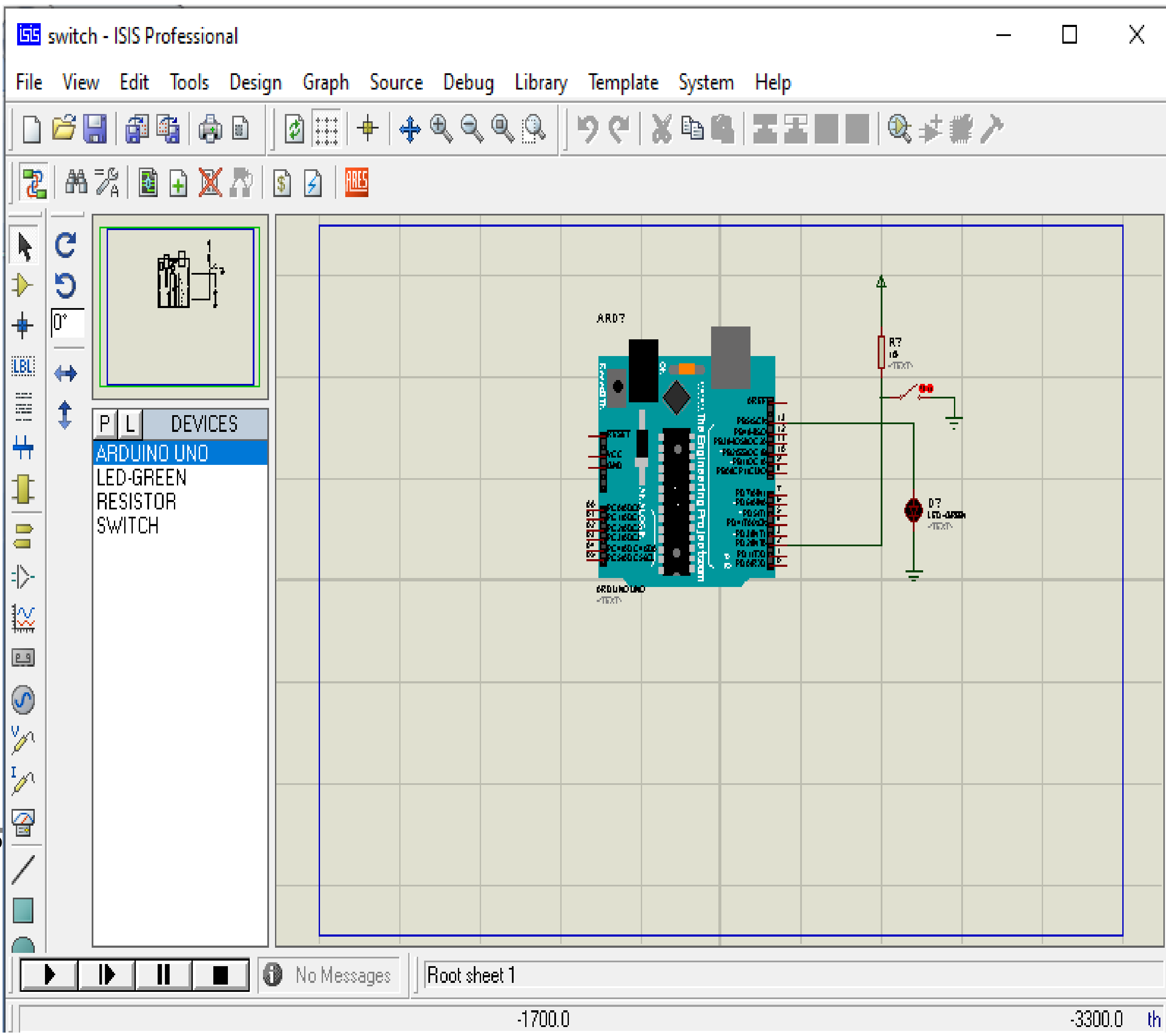
```
  digitalWrite(led, LOW);

  delay(200);

}
```



Experiment 4

Program-2 switch

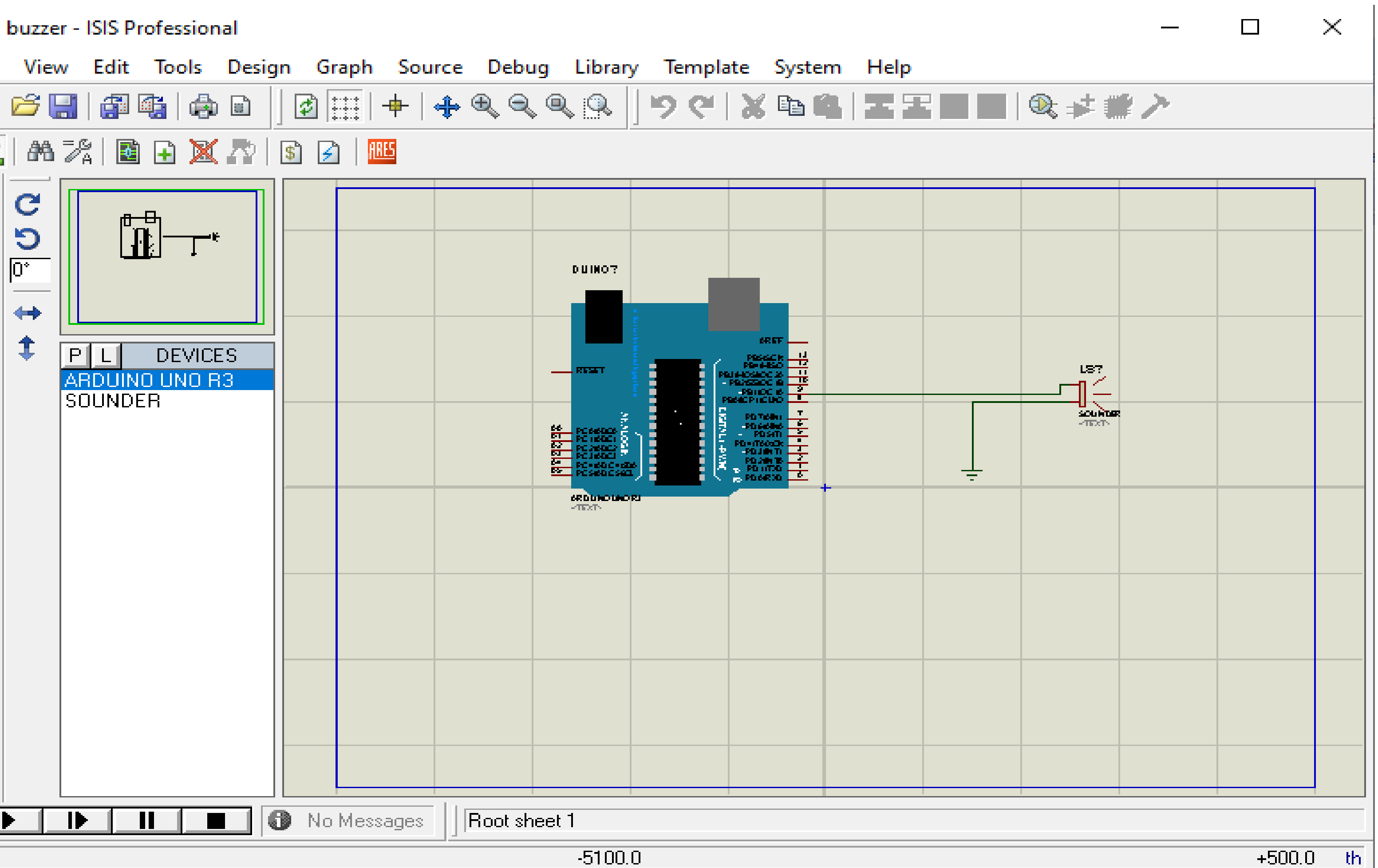
```
int switchPin = 2;
 int ledPin =  13;
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(switchPin, INPUT);
}
void loop()
{
  if ( digitalRead(switchPin) == HIGH  )
  {
    digitalWrite(ledPin, HIGH);
  }
    else
  {
    digitalWrite(ledPin, LOW);
  }
}
```


switch - ISIS Professional
File View Edit Tools Design Graph Source Debug Library Template System Help
P L DEVICES
ARDUINO UNO
LED-GREEN
RESISTOR
SWITCH
No Messages
Root sheet 1
-1700.0
-3300.0
th

P

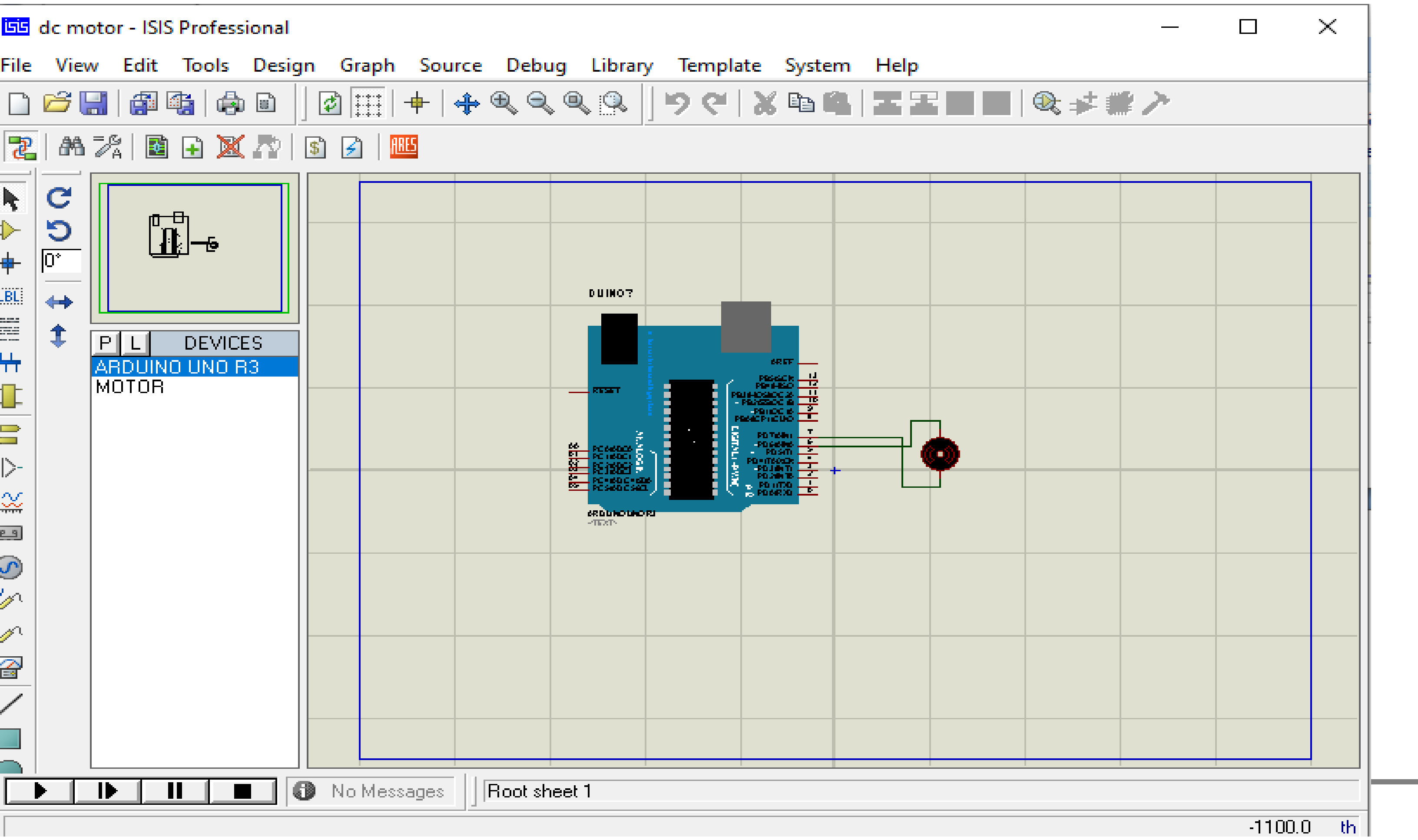
Program-4 ambulance

```
int buzzer = 9;
void setup()
{
 pinMode (buzzer ,OUTPUT);
}
void loop()
{
 for (int  i=0 ;      i <= 255;      i= i+ 5)
  {
    analogWrite(buzzer, i);
delay(30);
  }
for (int i = 255;        i >= 0;      i=i-5)
  {
    analogWrite(buzzer, i);
    delay(30);
  }
}
```

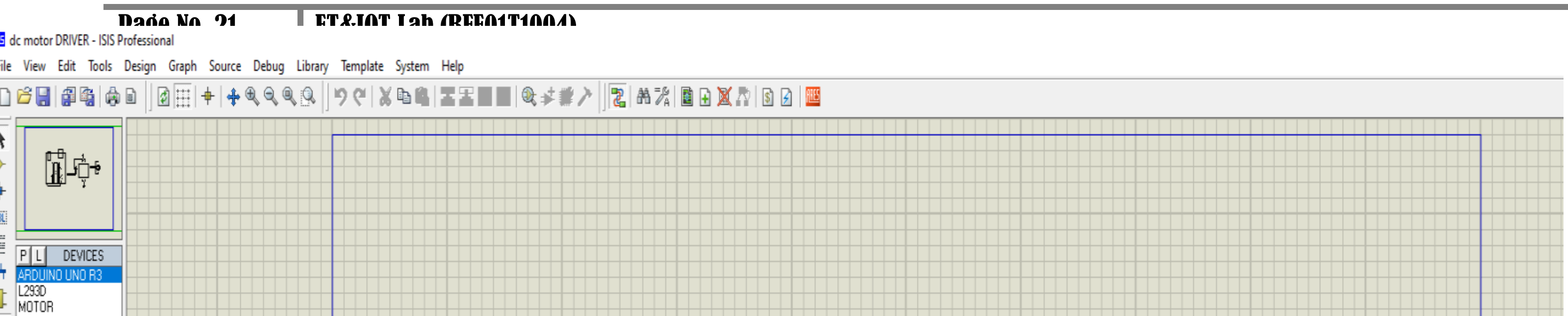

buzzer - ISIS Professional
View Edit Tools Design Graph Source Debug Library Template System Help
0°
P L DEVICES
ARDUINO UNO R3
SOUNDER
RESET
AREF
SOUNDER
No Messages
Root sheet 1
-5100.0
+500.0
th

Program-5 dc motor with fan

```
int motorpostitive =6;

int motornegative = 7;

void setup() {

 pinMode(motorpostitive ,OUTPUT );

 pinMode(motornegative ,OUTPUT );

}

void loop()

{

    digitalWrite (motorpostitive ,HIGH);

     digitalWrite (motornegative ,LOW);

      delay(500);

    digitalWrite (motorpostitive ,LOW);

      digitalWrite (motornegative ,HIGH);

      delay(500);

    }
```


dc motor - ISIS Professional
File View Edit Tools Design Graph Source Debug Library Template System Help
P L DEVICES
ARDUINO UNO R3
MOTOR
No Messages
Root sheet 1
-1100.0 th

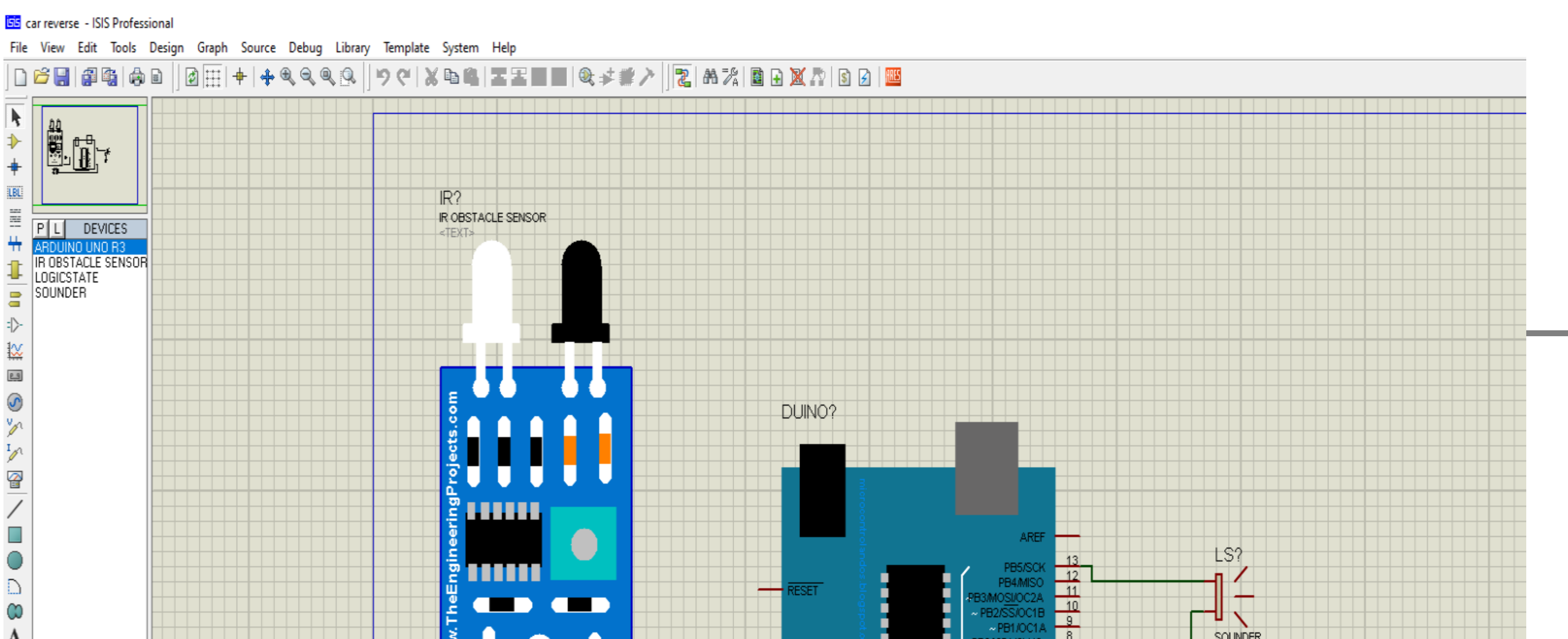
```
int motorpostitive =6;

int motornegative = 7;

void setup() {

 pinMode(motorpostitive ,OUTPUT );

 pinMode(motornegative ,OUTPUT );

}

void loop()

{

  digitalWrite (motorpostitive ,HIGH);

      digitalWrite (motornegative ,LOW);

      delay(500);

   digitalWrite (motorpostitive ,LOW);

      digitalWrite (motornegative ,HIGH);

      delay(500);

    }
```

Program-6 car reverse sensor

```
int irsensor = 2;

 int buzzer =  13;

void setup() {

  pinMode(irsensor, INPUT);
```

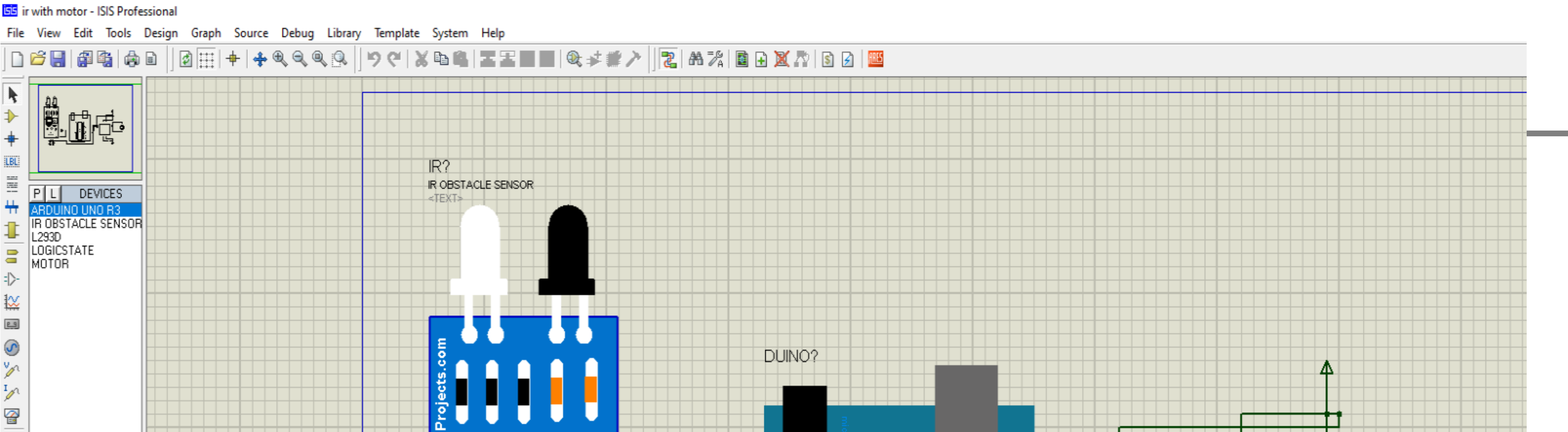
```
  pinMode(buzzer, OUTPUT);
}
void loop()
{
 if ( digitalRead(irsensor) == HIGH  )
  {
    digitalWrite(buzzer, HIGH);
  }
 if ( digitalRead(irsensor) == LOW  )
  {
    digitalWrite(buzzer, LOW);
  }
}
```



Program-7 ir with motor

```
int  irsensor  =5;

int motorpostitive =6;

int motornegative = 7;

void setup()
```

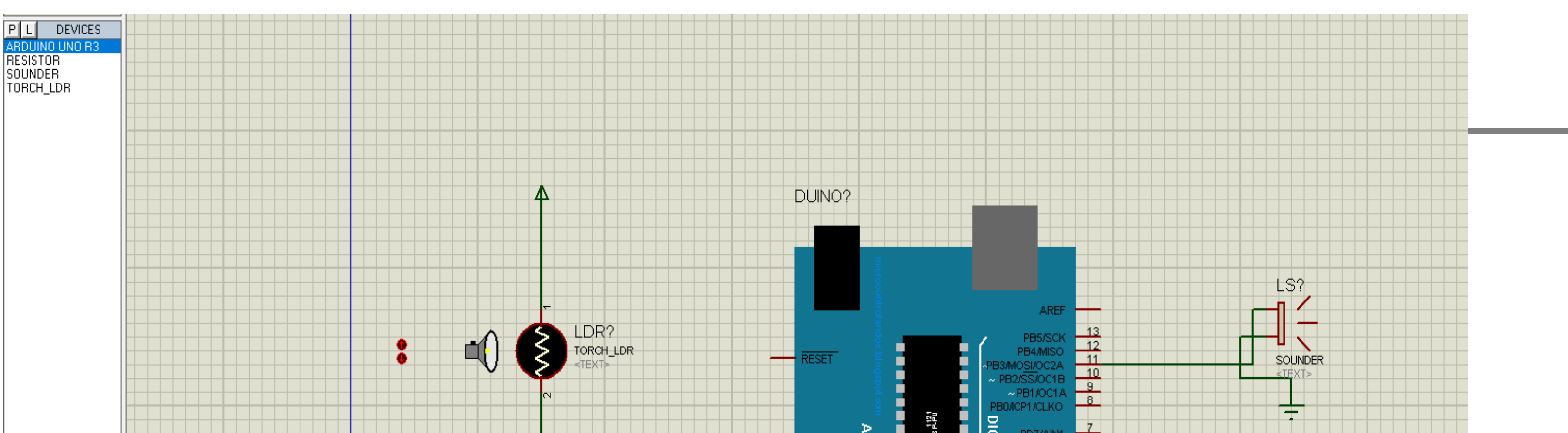
```
 {
 pinMode( irsensor ,INPUT );
 pinMode(motorpostitive ,OUTPUT );
 pinMode(motornegative ,OUTPUT );
}
void loop()
{
   if (digitalRead(irsensor)== HIGH)
   {
     digitalWrite (motorpostitive ,HIGH);
     digitalWrite (motornegative ,LOW);
   }
  if (digitalRead(irsensor)== LOW)
   {
     digitalWrite (motorpostitive ,LOW);
     digitalWrite (motornegative ,LOW);}}
```



Program-8 Ldr

```
int buzzer = 11;

void setup()

{ pinMode(buzzer,OUTPUT);

}
```

```
void loop()
 {
  int LDR = analogRead(A0);
   delay(500);
   if(LDR > 512)
  {
   digitalWrite(buzzer, HIGH);
   delay(10);
  }
  else
   {
    digitalWrite(buzzer, LOW);
    delay(10);
   }
 }
```

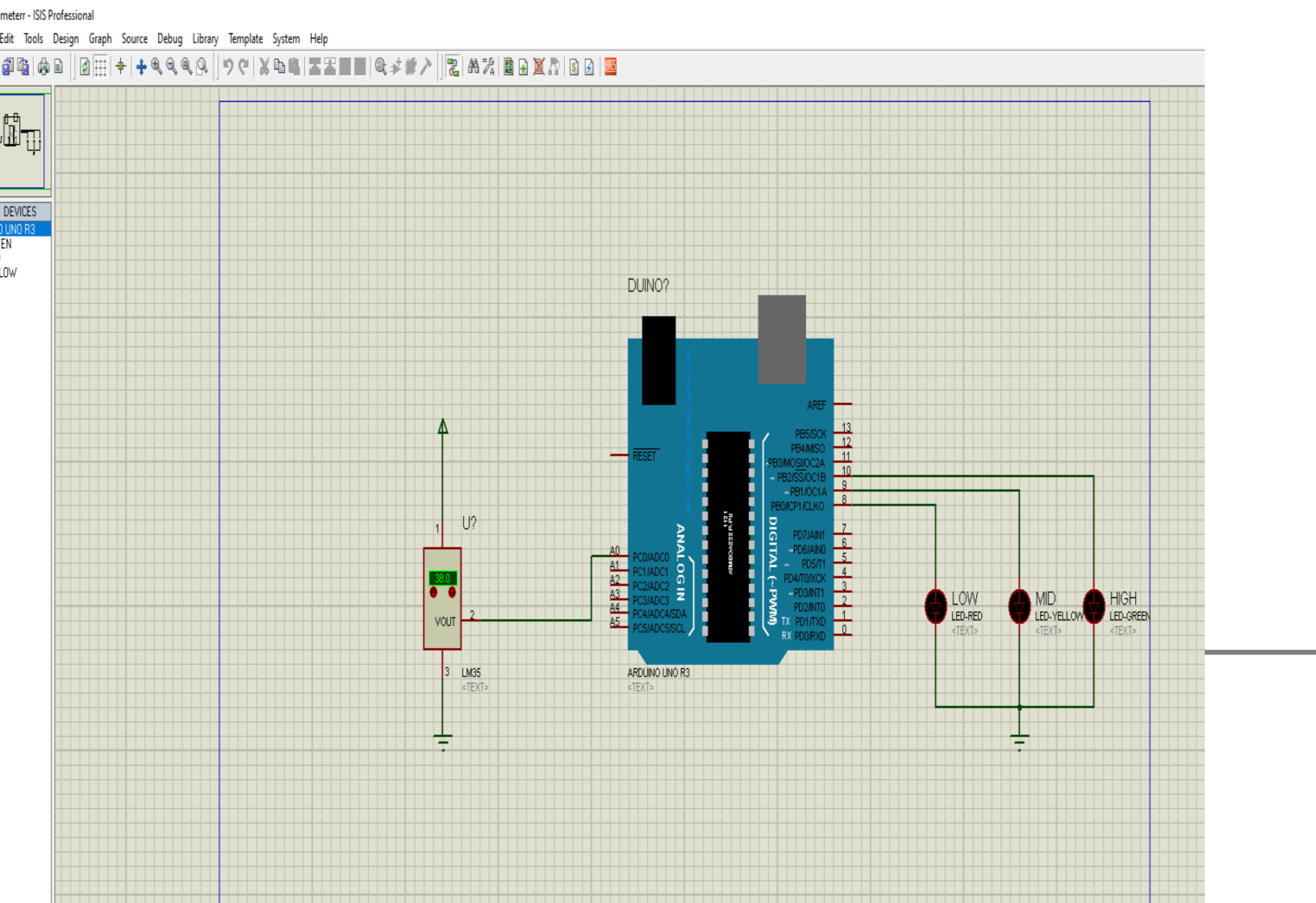


Program-9 temp- love o meter

```
int red = 8;

int yellow = 9;

int green = 10;

int tempsensor= A0;
```

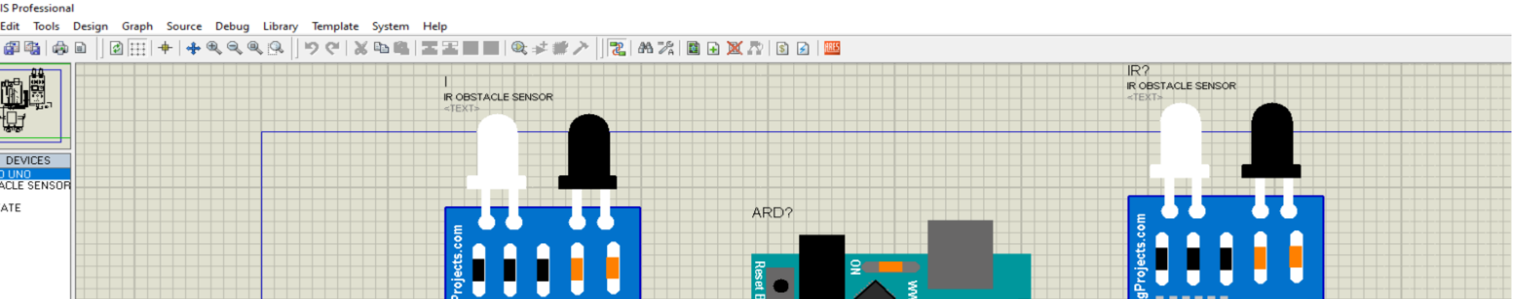
```
void setup()
{
  pinMode(red,OUTPUT);
   pinMode(yellow,OUTPUT);
    pinMode(green,OUTPUT);
  pinMode(tempsensor,INPUT);
}
void loop()
 {
  int temp = analogRead(tempsensor);
  if(temp < 100 &&  temp >1 )
  {
   digitalWrite(red, HIGH);
   digitalWrite(yellow, LOW);
    digitalWrite(green, LOW);
  }
 if(temp < 200  && temp >101)
 {
   digitalWrite(red, LOW);
   digitalWrite(yellow, HIGH);
    digitalWrite(green, LOW);
```

```
}
if(temp <300  &&   temp >201)
{
 digitalWrite(red, LOW);
 digitalWrite(yellow, LOW);
  digitalWrite(green, HIGH);
}
}
```



```
int leftmotornegative= 9;

int rightmotorpositive =10;

int rightmotornegative = 11;

void setup() {

 pinMode(leftmotorpositive, OUTPUT);

  pinMode(leftmotornegative, OUTPUT);

  pinMode(rightmotorpositive, OUTPUT);

  pinMode(rightmotornegative, OUTPUT);

  pinMode(leftsensor, INPUT);

  pinMode(rightsensor, INPUT);

}

void loop() {

 if (digitalRead(leftsensor)== HIGH && digitalRead(rightsensor) == HIGH)

  {

  digitalWrite(leftmotorpositive, HIGH);

    digitalWrite(leftmotornegative, LOW);

    digitalWrite(rightmotorpositive, HIGH);

    digitalWrite(rightmotornegative, LOW);

}

  if (digitalRead(leftsensor)== LOW && digitalRead(rightsensor) == HIGH)

  {
```

```
   digitalWrite(leftmotorpositive, LOW);
  digitalWrite(leftmotornegative, LOW);
  digitalWrite(rightmotorpositive, HIGH);
  digitalWrite(rightmotornegative, LOW);

 }
 if (digitalRead(leftsensor)== HIGH && digitalRead(rightsensor) == LOW)
 {
  digitalWrite(leftmotorpositive, HIGH);
  digitalWrite(leftmotornegative, LOW);
  digitalWrite(rightmotorpositive, LOW);
  digitalWrite(rightmotornegative, LOW);
}
 if (digitalRead(leftsensor)== LOW && digitalRead(rightsensor) == LOW)
 {
 digitalWrite(leftmotorpositive, LOW);
  digitalWrite(leftmotornegative, LOW);
  digitalWrite(rightmotorpositive, LOW);
  digitalWrite(rightmotornegative, LOW);
 }
 }
```



Do and Don'ts

- Avoid contact with energized electrical circuits.

- Powered equipment can be hot! Use caution when handling equipment after it has been operating.
- Select proper type of supply (i.e. a. c. or d. c.) and range of meters.
- All the connections should be tight.
- Never exceed the permissible values of current, voltage, and apparatus, wire, load, etc.
- If water or a chemical is spilled onto equipment, shut off power at the main switch or circuit breaker and unplug the equipment.
- Be sure you understand the function and wiring of an instrument before using it in a circuit.